

Multi-digit



Opgave

Met het gegeven cijfer 3 en de bewerkingen + en \times kan je getallen maken, bijvoorbeeld:

$$\begin{array}{lll} 9 & = & 3 \times 3 & \text{[kost = 2]} \\ 6 & = & 3 + 3 & \text{[kost = 2]} \\ 36 & = & 33 + 3 & \text{[kost = 3]} \\ 72 & = & (3 + 3) \times (3 + 3 \times 3) & \text{[kost = 5]} \end{array}$$

De kost is dan het aantal keer dat het cijfer 3 voorkomt in de rekenkundige uitdrukking. Je ziet dat je het gegeven cijfer gewoon een aantal keer achter elkaar mag plaatsen, met de gebruikelijke betekenis van de waarde ervan. Dus 3333333 kost 7.

De bedoeling is om altijd een uitdrukking te bekommen met de kleinste kost, dus al is 72 ook gelijk aan $(33 + 3) + (33 + 3)$, dat is niet een oplossing die we aanvaardden, want die heeft kost = 6.

Je krijgt mogelijk meer dan één cijfer om nieuwe getallen te maken, bijvoorbeeld de cijfers 2 en 7:

$$729 = (2 + 7) \times (2 + (2 + 77)) \quad \text{[kost 6]}$$

Alle gegeven cijfers liggen strikt tussen 0 en 10, maar het te berekenen getal kan heel groot zijn: zo heeft 100,000 bijvoorbeeld een expressie met de cijfers 2 en 7 met minimale kost 11.

Als de gegeven cijfers 2 en 7 zijn, dan mag je wel 22 en 222 ... of 77 en 777 ... gebruiken, maar niet (bijvoorbeeld) 72: je kan enkel een cijfer herhalen.

Invoer

De eerste regel van de invoer bestaat uit het aantal testgevallen $1 \leq n \leq 1000$. Daarna komen n regels, één per testgeval. Deze regels bestaan uit een aantal getallen gescheiden door spaties. Het eerste getal $k \in [1, 9]$ stelt het aantal gegeven cijfers voor. De volgende k getallen $c_1, \dots, c_k \in [1, 9]$ zijn onderling verschillende cijfers in oplopende volgorde. Het laatste getal op de regel is $G \in [1, 100,000]$, het getal dat moet samengesteld worden op basis van de gegeven cijfers c_1, \dots, c_k en de bewerkingen \times en $+$.

Uitvoer

De uitvoer bestaat uit n regels, één regel per testgeval. De regel van een testgeval bevat één enkel geheel getal dat de minimale kost om G te berekenen op basis van de gegeven cijfers voorstelt. Indien G niet kan berekend worden op basis van de cijfers, dan geef je als uitvoer 0.

Voorbeeld

Invoer

```
6
1 3 3
1 3 10
1 3 12
1 3 24
2 2 3 10
2 2 3 24
```

Uitvoer

Enkel het eerste getal op elke regel maakt deel uit van de uitvoer, je hoeft de expressie dus *niet* uit te schrijven.

```
1      % 3
0
3      % 3 x 3 + 3
5      % (3 + 3) x 3 + 3 + 3
3      % 2 x (2 + 3)
3      % 22 + 2
```